

```

--=====
-- SCRIPT ATUALIZADO (VERSÃO FINAL - SISTEMA DE QUESTÕES/RESPOSTAS)
--=====

-- 1. LIMPEZA (Dropa tudo para recriar do zero)
DROP TABLE IF EXISTS Resposta CASCADE;
DROP TABLE IF EXISTS Prova_Questao CASCADE;
DROP TABLE IF EXISTS Questao CASCADE;
DROP TABLE IF EXISTS Trabalho CASCADE;
DROP TABLE IF EXISTS Avaliacao CASCADE; -- Antiga tabela de prova
DROP TABLE IF EXISTS Nota_Criterio CASCADE; -- Tabela antiga (removida)
DROP TABLE IF EXISTS Criterio CASCADE; -- Tabela antiga (removida)
DROP TABLE IF EXISTS Est_part_discp CASCADE;
DROP TABLE IF EXISTS Disciplina CASCADE;
DROP TABLE IF EXISTS Professor CASCADE;
DROP TABLE IF EXISTS Estudante CASCADE;
DROP TABLE IF EXISTS Usuario CASCADE;

--=====
-- 2. CRIAÇÃO DAS TABELAS
--=====

-- 1. Tabela Base de Usuários
CREATE TABLE Usuario (
    id_usuario SERIAL PRIMARY KEY,
    pnome VARCHAR(50) NOT NULL,
    snome VARCHAR(50), -- Deixei opcional pois as vezes não vinha no cadastro simples
    email VARCHAR(100) UNIQUE NOT NULL,
    senha VARCHAR(255) NOT NULL
);

-- 2. Especialização: Estudante
CREATE TABLE Estudante (
    id_usuario INT PRIMARY KEY,
    matricula_academica VARCHAR(50),
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario) ON DELETE CASCADE
);

-- 3. Especialização: Professor
CREATE TABLE Professor (
    id_usuario INT PRIMARY KEY,
    departamento VARCHAR(100),
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario) ON DELETE CASCADE
);

```

```

-- 4. Tabela de Disciplina
CREATE TABLE Disciplina (
    id_disciplina SERIAL PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    id_professor INT NOT NULL, -- Professor responsável
    FOREIGN KEY (id_professor) REFERENCES Professor(id_usuario)
);

-- 5. Tabela de Matrícula (Estudante <-> Disciplina)
CREATE TABLE Est_part_discp (
    id_usuario INT NOT NULL, -- Aluno
    id_disciplina INT NOT NULL,
    PRIMARY KEY (id_usuario, id_disciplina),
    FOREIGN KEY (id_usuario) REFERENCES Estudante(id_usuario) ON DELETE CASCADE,
    FOREIGN KEY (id_disciplina) REFERENCES Disciplina(id_disciplina) ON DELETE CASCADE
);

-- 6. Tabela de Avaliação (A Prova)
CREATE TABLE Avaliacao (
    id_avaliacao SERIAL PRIMARY KEY,
    id_disciplina INT NOT NULL,
    crie_aval_prof_id INT NOT NULL, -- Professor criador
    cod_avaliacao VARCHAR(50) NOT NULL, -- Ex: AP1, AP2
    data_publicacao DATE DEFAULT CURRENT_DATE,
    prazo DATE,
    max_questoes INT DEFAULT 10, -- COLUNA NOVA: Define limite de questões

    FOREIGN KEY (id_disciplina) REFERENCES Disciplina(id_disciplina) ON DELETE CASCADE,
    FOREIGN KEY (crie_aval_prof_id) REFERENCES Professor(id_usuario),
    UNIQUE(id_disciplina, cod_avaliacao)
);

-- 7. Tabela de Questão (Banco de Questões) - [NOVA ESTRUTURA]
CREATE TABLE Questao (
    id_questao SERIAL PRIMARY KEY,
    enunciado TEXT NOT NULL,
    tipo VARCHAR(50) DEFAULT 'Discursiva', -- Ex: Discursiva, Multipla Escolha
    id_disciplina INT REFERENCES Disciplina(id_disciplina) ON DELETE CASCADE
    -- A questão pertence à disciplina, permitindo reuso em várias provas
);

-- 8. Tabela de Ligação (Montagem da Prova) - [NOVA TABELA]
CREATE TABLE Prova_Questao (
    id_avaliacao INT NOT NULL,

```

```
    id_questao INT NOT NULL,  
    PRIMARY KEY (id_avaliacao, id_questao),  
    FOREIGN KEY (id_avaliacao) REFERENCES Avaliacao(id_avaliacao) ON DELETE  
CASCADE,  
    FOREIGN KEY (id_questao) REFERENCES Questao(id_questao) ON DELETE  
CASCADE  
);
```

-- 9. Tabela de Submissão de Arquivo (PDF)

```
CREATE TABLE Trabalho (  
    id_trabalho SERIAL PRIMARY KEY,  
    id_avaliacao INT NOT NULL,  
    id_usuario_est INT NOT NULL, -- Aluno  
    data_envio TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    arquivo_url TEXT, -- Link do PDF  
    status VARCHAR(20) DEFAULT 'Enviado',  
  
    FOREIGN KEY (id_usuario_est) REFERENCES Estudante(id_usuario),  
    FOREIGN KEY (id_avaliacao) REFERENCES Avaliacao(id_avaliacao) ON DELETE  
CASCADE,  
    UNIQUE(id_avaliacao, id_usuario_est)  
);
```

-- 10. Tabela de Respostas e Notas (Correção) - [NOVA TABELA]

```
CREATE TABLE Resposta (  
    id_resposta SERIAL PRIMARY KEY,  
    id_estudante INT REFERENCES Estudante(id_usuario),  
    id_avaliacao INT REFERENCES Avaliacao(id_avaliacao) ON DELETE CASCADE,  
    id_questao INT REFERENCES Questao(id_questao),  
  
    resposta_texto TEXT, -- O que o aluno escreveu  
    nota_obtida DECIMAL(5,2) DEFAULT 0.0, -- Nota dada pelo professor (0 a 10)  
    data_envio TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  
    -- Garante que um aluno só responde uma vez a mesma questão na mesma prova  
    UNIQUE(id_estudante, id_avaliacao, id_questao)  
);
```